

SoftSide™ Selections



A product of **SoftSide** Publications, Inc.
10 Northern Blvd., Amherst, NH 03031

SoftSideTM Selections

Index

Microtext 2.0
Microtext 2.0

FRONT RUNNER: Microtext 2.0

by Jon Voskuil

Modifications and enhancements by the
SoftSide programming staff.

Here is an easy-to-use BASIC word
processor with edit, printout and search
commands, for the whole family **2**



DV BONUS PROGRAM: Life II

by Leo Christopherson

Documentation by Rich Bouchard

Watch cells get born, multiply, form col-
onies and die in this realistic simulation
game **13**



King II

by John F. Gabriel

Royal ways and whims determine the
moves in this political war game of epic
proportions **16**



ADVENTURE DISK and CASSETTE BONUS: **SoftSide Adventure Series: The Arabian Nights** by Peter Kirsch

Brave the perils of ancient Baghdad to
save the fair Princess in distress **28**

General Information **and System Requirements** **32**

This booklet prepared by the staff of SoftSide Publications, Inc.,
Entire contents copyright © 1983 by SoftSide Publications, Inc., 10
Northern Blvd., Northwood Executive Park, Amherst, NH 03031.

TRS-80® is a trademark of the Tandy Corporation.





by Jon Voskuil

Modifications and enhancements by the **SoftSide** programming staff.

Microtext 2.0 is a word processor for a 16K TRS-80® Model I or III (32K with disk).

Microtext 2.0 displays a mostly blank screen with an instruction summary line at the top or bottom. You can start typing, or load a previously saved file from disk or tape. To capitalize characters on the TRS-80 Model I version, place an "@" symbol before the letter. Use two ("@@") to capitalize the entire word. Normal use of the Model I's shift key inserts the "@" in front of individual characters. TRS-80 Model III users, of course, can work in true mixed case. TRS-80 Model I users with a lower-case modification and driver installed should delete line 470 to use true mixed case. Naturally, you should check your typing with *SWAT* before making any changes. In addition, pressing SHIFT-left-arrow erases the line you are currently typing.

Primary Commands

Use the CLEAR key to access editor and system functions. Pressing and releasing CLEAR, followed by S, L, R, P, or E enters the save, load, review, printout, or edit functions. Although not mentioned in the command summary on the screen, pressing CLEAR-Q will quit the program.

Saving and loading files is simply a matter of answering the questions about the medium to be used (tape or disk) and, if disk, the file name. Once you have entered a file name, it becomes the default until you specify another one or exit the program: just press Enter when asked for the file name. This simplifies repeated saves during entry of a long document.

The review function causes the computer to return to the beginning of the text, and scroll through it to the end. Press the space bar to pause. Press the space bar again to display one or more lines. Use Enter to continue the scrolling. E takes you to the editing mode.

Editing

To move the cursor up and down, use the up- and down-arrow keys. CLEAR exits to the review mode. D deletes the line at the cursor. X deletes everything from the cursor to the end of the text. Enter allows you to edit the line at the cursor. F finds a string in your text.

If you choose to edit a line, the screen clears, and displays several lines of text with a gap in the middle. The cursor is at the beginning of the line, and you can proceed to type in a new one. It can be shorter than the original, or occupy multiple screen lines. To retain any part of the original line, use the right-arrow key to retype that portion automatically. Whatever you type in replaces the entire line. When you finish entering the new text, press CLEAR-F (not Enter, unless you want a carriage return in the text itself). The computer checks to see if the text lines need to be rearranged, and returns you to the review mode.

Find

The find function, accessed by pressing F from the edit mode, asks you what string to search for. If you have already done a search, Enter defaults to the previous string. The program looks for your string, beginning with the line the cursor is on, and puts the cursor on the first line in which your string occurs. Since a multiple-word string may be broken between lines, searching for single words is more reliable. The find routine places you in the edit mode, or displays a message to inform you the string was not found.

Printing

The printout function sends your text to a printer, after selecting margins, line spacing, and the case-conversion option. Pressing Enter in response to the offered options selects the default value indicated. The case conversion option prints the text *exactly* as it appears on the Model I's screen (all upper case and with the embedded @ symbols), or converts it (a bit sluggishly, using a BASIC routine) to normal mixed case for a final printout.

Variables

B\$: CHR\$(8), used to erase one character to the left of the cursor.
BK: Equals 8, the ASCII code of the backspace character.
C: The ASCII code of the character mode recently typed.
C\$: Used to hold either a single character from the keyboard or a character in a line of text.
C1: Temporary storage for CH.
CC: The ASCII code of the character underneath the cursor.
CH: The number of characters that have been typed in the current line.

CL: Equals 31, the character code for "clear to end of screen."
CR: Equals -1 if the current line ends with a carriage return.
CR\$: Equals CHR\$(140), the graphic block used to represent a carriage return.
CU\$: Equals CHR\$(95), the underscore character used as a cursor.
CV: Equals -1 if the program is to convert the text to upper and lower case using the "@" convention described above.

MICROTEXT 2.0

CX\$: Used to hold a single character from a line of text.
ED: Equals -1 if the program is in the edit mode.
EL: The number of the line being edited.
F\$: File name.
F1\$: Previous file name (if any).
I: General loop variable.
IT: Used in screen display routine to determine which line is to be the first displayed.
J, J1, K: General loop variables.
L: Line length.
L\$: Temporary storage of a text line.
L\$(*): Main string array to hold lines of text.
L1, L2: Used to determine the first and last lines to be displayed on the screen during an editing session.
LC: Used in printer output routine when using "@" conversion to upper and lower case to determine whether a character should be printed in upper or lower case.
LI: Stored the position of the printer on a page.
LL: Line length.
LM: Left margin for printed output.
LN: Number of lines of text in the array L\$(*).*
LO: Caps lock flag for print routine.
LP: Length of P\$.
LS: Line spacing.
LW: Maximum number of characters per screen line.

MD: Equals 3 if the program is running on a Model III, or 1 if the program is running on a Model I.
NL: Pointer used in edit routine.
NN: Pointer used to display lines in edit routine.
P, PP: Hold PRINT@ screen locations.
P\$: Text line to be printed.
QS: Contains the edit mode prompt line.
RE: Equals 140, the character code for the graphics block used to represent carriage returns.
RM: Right margin.
RT: Equals 13, the ASCII code of the Enter key.
S\$: Search string.
S1: Temporary storage for SL.
S1\$: Storage for the previous search string.
SH: Equals -1 if we are using a Model I and are placing "@" characters before characters that are entered using the shift key.
SL: Last character position at which a space occurred.
SP: Equals 32, the ASCII code for the space character.
SS: Number of characters to erase when word-wrapping.
ST: Equals -1 if we are in the review mode and are stepping through the file line by line with the spacebar.
TL: Used to manage string position in the edit mode.
V1: Vertical screen position.
X, X\$: General usage.
Z: Timing loop variable.

ast.
hough it so
ow process, it wa
by 15th m

```

SS SS SS SS SS SS SS SS SS SS SS
SS
SS      TRS-80 BASIC      SS
SS      'Microtext 2.0'   SS
SS      Author: Jon R. Voskuil  SS
SS      Copyright © 1983  SS
SS      SoftSide Publications, Inc SS
SS
SS
SS SS SS SS SS SS SS SS SS SS SS

```

**If you don't wish to type this program, it
is available on Issue #42 SoftSide CV
and DV.**

Branch to start of program.

```
100 CLEAR50:GOTO200
```

Subroutine to input a single character from the keyboard.

```

110 C=PEEK(14400):IFC=0THEN140
120 IFC=8THENC=91ELSEIFC=16THENC=10ELSEIFC=32THENC=8ELSEIFC=64TH
ENC=9ELSEIFC=128THENC=32ELSE140
130 RETURN
140 C=PEEK(14463):IFC=0THEN140ELSEC$=INKEY$:IFC$=""THEN140
150 C=ASC(C$):RETURN
160 GOSUB110:IFC=97ANDC<=122THENC=C-32
170 RETURN

```

Subroutine to wait for the ENTER key to be pressed.

```
180 IFINKEY$<>CHR$(13)THEN180ELSEReturn
```

Display title page.

```

200 CLS:PRINTCHR$(23)
210 PRINT@198,"M I C R O T E X T   2 . 0"
220 PRINT@398,"BY JON R. VOSKUIL":PRINT@512,"MODIFICATIONS BY RI
CH BOUCHARD"
230 PRINT@652,"COPYRIGHT (C)  1983":PRINT@708,"SOFTSIDE PUBLICAT
IONS, INC."
240 FORZ=1TO800:IFINKEY$=""THENNEXTZ
250 GOTO400

```

Lower-case print routine.

```

300 IFPP$=""THEN370
310 LOK=0:LC=-1:FORK=1TOLEN(PP$):CC=ASC(MID$(PP$,K,1)):IFCC=32TH
ENLOK=0
320 IFCC<>64THEN350
330 IFLC=0THENLOK=-1
340 LC=0:GOTO360
350 LPRINTCHR$(CC-32*(LCANDCC>64ANDCC<91)*-(NOTLOK));LC=-1
360 NEXTK
370 RETURN

```


MICROTEXT 2.0

Initialization.

```

400 CLS
410 IF MEM>35000 THEN CLEAR 32000 ELSE CLEAR MEM-3000
420 DEFINT A-Z
430 DIM C, B$, CH, LN, CC, CU$, SL, LW, LN
440 DIM L$(FRE(A$)/50)
450 IF PEEK(293)=73 THEN MD=3 ELSE MD=1
460 IF MD=3 THEN POKE 16916, 2
470 SH=0: IF MD=1 THEN SH=-1
480 ON ERROR GOTO 10000
490 BK=8: RT=13: SP=32: CL=31: B$=CHR$(8): RE=140: CR$=CHR$(RE)
500 CH=1
510 CU$=CHR$(95)
520 LN=1
530 LW=62
540 P=128

```

Main text input loop.

```

550 IF E THEN 570 ELSE PRINT @, CHR$(30); "SAVE: CLR-S  LOAD: CLR-L  R
    EVW: CLR-R  EDIT: CLR-E  PRINT: CLR-P"
560 PRINT @64, STRING$(63, "-");
570 PRINT @P, "";
600 CC=PEEK(PEEK(16416)+PEEK(16417)*256): PRINT CU$; IF CC<32 THEN CC
    =CC+64
610 GOSUB 110
620 IF C<32 THEN 690
630 PRINT B$; IF SH AND C=97 AND C<=122 THEN CH=CH+1: L$(LN)=L$(LN)+"@":
    PRINT "@"; C=C-32
640 CH=CH+1: IF CH=LW AND C<>SP AND C<>RE THEN GOSUB 1000

```

Add character to current string and display on screen.

```

650 L$(LN)=L$(LN)+CHR$(C)
660 PRINT CHR$(C); IF C<>32 AND C<>RE THEN 600
670 IF C=32 THEN SL=CH: IF CH=LW THEN LN=LN+1: L$(LN)="" : CH=1: SL=0: PRINT
    : GOTO 600 ELSE 600
680 PRINT CHR$(30): LN=LN+1: L$(LN)="" : SL=0: CH=1: GOTO 600

```

Routine to handle right-arrow, carriage return, delete, shifted left-arrow and clear.

```

690 IF C=9 THEN CC=60: GOTO 630
700 IF C=RT THEN CC=RE: GOTO 630
710 IF C=BK THEN IF C<2 THEN 610 ELSE PRINT B$; B$; CH=CH-1: L$(LN)=LEFT$(
    L$(LN), LEN(L$(LN))-1): GOTO 600
720 IF C=24 THEN PRINT CHR$(29); CHR$(30); L$(LN)="" : GOTO 600
730 IF C=CL THEN 1110
740 GOTO 610

```

Word-wrap line.

```

1000 IF SL=0 THEN PRINT: GOTO 1060
1010 SS=LW-SL: FOR J=1 TO SS: PRINT B$; NEXT J
1020 PRINT STRING$(SS, 32); PRINT

```



```

1030 IFSL=LW-1THEN1060
1040 L$(LN+1)=RIGHT$(L$(LN),LW-1-SL)
1050 L$(LN)=LEFT$(L$(LN),SL-1)
1060 LN=LN+1
1070 PRINTL$(LN);CU$;
1080 CH=LEN(L$(LN))+2
1090 SL=0
1100 RETURN

```

Get a character following a CLEAR character.

```

1110 P=PEEK(16416)+PEEK(16417)*256-15360
1120 POKEP+15359,143
1130 GOSUB160
1140 POKEP+15359,32
1150 IFC=70ANDDTHENED=0:RETURN
1160 IFDTHEN1230
1170 IFC=82THENGOSUB1240:GOTO550
1180 IFC=83THENGOSUB1430:GOTO550
1190 IFC=76THENGOSUB1680:GOTO550
1200 IFC=81THEN11000
1210 IFC=80THENGOSUB2000:GOTO550
1220 IFC=69ANDLN>1THENI=LN-1:PP=P:GOSUB2630:GOSUB1240:GOTO550
1230 P=P-1:GOTO550

```

Review routine.

```

1240 CLS:PRINT@,CHR$(30);TAB(21);"PRESS ANY KEY TO PAUSE":PRINT
STRING$(63,"-")
1250 IFLN=1THEN1400
1260 FORI=1TOLN-1
1270 FORZ=1TO20:NEXTZ
1280 PRINTL$(I)
1290 IFINKEY$=""ANDNOTSTTHEN1390
1300 ST=0
1310 PP=PEEK(16416)+PEEK(16417)*256-15360
1320 PRINT@,CHR$(30);"  ENTER: CONTINUE          SPACEBAR: STEP 1
LINE      E:EDIT      ";STRING$(63,"-");
1330 GOSUB160
1340 IFC=69THENGOSUB2630:GOTO1240
1350 IFC=13THEN1380
1360 IFC<>32THEN1320
1370 ST=-1
1380 PRINT@PP,"";
1390 NEXTI
1400 PRINTL$(LN);
1410 P=PEEK(16416)+PEEK(16417)*256-15360
1420 RETURN
Save routine.
1430 PRINT@64,STRING$(63,"-");:PRINT@,STRING$(63,32);:PRINT@,"
SAVE TO TAPE OR DISK? (T/D/CLEAR) ";
1440 GOSUB160
1450 IFC=CLTHEN1670

```

MICROTEXT 2.0

```
1460 L$(0)=STR$(LN)+STR$(10000+SL*100+CH)
1470 PRINT@0,CHR$(30);
1480 IFC=84THEN1620
1490 IFC=68THEN1430
1500 F1$=F$
1510 PRINT@0,CHR$(30);:LINEINPUT"FILE NAME: ";F$
1520 IFF$=""ANDF1$=""THEN1510
1530 IFF$=""THENF$=F1$
1540 PRINT@0,CHR$(30);"INSERT DISK AND PRESS ENTER.":PRINTSTRING
$(63,"-");
1550 GOSUB180
1560 OPEN"D",1,F$
1570 FORI=0TOLN
1580 PRINT#1,L$(I)
1590 NEXTI
1600 CLOSE
1610 GOTO1670
1620 PRINT@0,CHR$(30);"START RECORDER AND PRESS ENTER.";
1630 GOSUB180
1640 FORI=0TOLN
1650 PRINT#-1,CHR$(34);L$(I);CHR$(34)
1660 NEXTI
1670 P=P-1:RETURN
Load routine.
1680 PRINT@64,STRING$(63,"-");:PRINT@0,CHR$(30);"LOAD FROM TAPE
OR DISK? (T/D/CLEAR) ";
1690 GOSUB160
1700 IFC=CLTHEN1940
1710 PRINT@0,STRING$(40,32);
1720 IFC=84THEN1880
1730 IFC<>68THEN1680
1740 F1$=F$
1750 PRINT@0,CHR$(30);:LINEINPUT"FILE NAME: ";F$
1760 IFF$=""ANDF1$=""THEN1750
1770 IFF$=""THENF$=F1$
1780 PRINT@0,CHR$(30);"INSERT DISK AND PRESS ENTER.":PRINTSTRING
$(63,"-");
1790 GOSUB180
1800 OPEN"I",1,F$
1810 INPUT#1,L$(0)
1820 GOSUB1950
1830 FORI=1TOLN
1840 LINEINPUT#1,L$(I)
1850 NEXTI
1860 CLOSE
1870 GOTO1940
1880 PRINT@0,CHR$(30);"START RECORDER AND PRESS ENTER.";
1890 GOSUB180
```



```

1900 INPUT#-1,L$(0):GOSUB1950
1910 FORI=1TOLN
1920 INPUT#-1,L$(I)
1930 NEXTI
1940 GOSUB1240:RETURN
1950 L$=L$(0):L=LEN(L$)
1960 CH=VAL(RIGHT$(L$,2)):L$=LEFT$(L$,L-2)
1970 SL=VAL(RIGHT$(L$,2)):L$=LEFT$(L$,L-5)
1980 LN=VAL(L$)
1990 RETURN
Print routine.
2000 PRINT@0,CHR$(31);:PRINT@192,,:LIN=0
2010 X$="10":INPUT"LEFT MARGIN (DEFAULT = 10) ";X$:LM=VAL(X$)
2020 PRINT:X$="70":INPUT"RIGHT MARGIN (DEFAULT = 70) ";X$:RM=VAL(X$)
2030 PRINT:X$="2":INPUT"LINE SPACING (DEFAULT = 2) ";X$:LS=VAL(X$)
2040 LL=RM-LM
2050 PRINT:X$="N":INPUT"CONVERT TO LOWERCASE, UNLESS PRECEDED BY
@ (DEFAULT = NO) ";X$:CV=0:IFLEFT$(X$,1)="Y"ORLEFT$(X$,1)=CHR$(
121)THENCV=-1
2060 CLS:LPRINT " ":P$="":CR=0:I=0
2070 I=I+1:P$=P$+L$(I)
2080 IFRIGHT$(P$,1)=CR$THENCV=-1:GOTO2100
2090 IFLEN(P$)<255-LWANDI<LNTHEN2070
2100 GOSUB2170:CR=0
2110 IFI<LNTHEN2070
2120 LPRINTTAB(LM);:IFCV=0THENLPRINTP$;:GOTO2140
2130 PP$=P$:GOSUB300
2140 LPRINT " "
2150 GOSUB1240
2160 RETURN
2170 L=LL
2180 IFLEN(P$)>LLTHEN2220
2190 IFNOTCRTHEN2320
2200 LP=LEN(P$):IFLP<2THENPP$="":P$="":GOTO2260
2210 PP$=LEFT$(P$,LP-1):P$="":GOTO2260
2220 C$=MID$(P$,L,1):IFC$=" "THEN2250
2230 L=L-1:IFL>0THEN2220
2240 L=LL
2250 PP$=LEFT$(P$,L):P$=RIGHT$(P$,LEN(P$)-L)
2260 LPRINTTAB(LM);:IFCV=0THENLPRINTPP$;:GOTO2280
2270 GOSUB300
2280 FORJ=1TOLS:LIN=LIN+1:LPRINT " :NEXTJ
2290 IFLIN>59THENFORJ=1TO66-LIN:LPRINT " :NEXTJ:LIN=0
2300 IFLEN(P$)>LLTHENL=LL:GOTO2220
2310 IFCRANDLEN(P$)>0THEN2200
2320 RETURN

```

MICROTEXT 2.0

Re-justification routine.

```

2330 CLS:PRINT@0,CHR$(30);"RE-JUSTIFYING TEXT. . ."
2340 LIN=EL:LN=LN-1
2350 P$="":CR=0:I=EL-1
2360 I=I+1:P$=P$+L$(I)
2370 IFRIGHT$(P$,I)=CR$THENCN=-1:GOTO2390
2380 IFLEN(P$)+LEN(L$(I+1))<256ANDI<LNTHEN2360
2390 GOSUB2490
2400 IFNOTCRTHEN2430
2410 X=I+1-LN:IFX=0THEN2480
2420 FORJ=I+1TOLN:L$(J-X)=L$(J):NEXTJ:LN=LN-X:GOTO2480
2430 IFI<LNTHEN2360
2440 L$(LIN)=P$
2450 CH=LEN(P$)+1:SL=LEN(P$)
2460 FORI=LIN+1TOLN:L$(I)="" :NEXTI
2470 LN=LIN
2480 RETURN
2490 L=LW
2500 IFLEN(P$)>LWTHEN2540
2510 IFNOTCRTHEN2620
2520 LP=LEN(P$)
2530 PP$=LEFT$(P$,LP):P$="":GOTO2580
2540 C$=MID$(P$,L,1):IFC$=" "THEN2570
2550 L=L-1:IFL>0THEN2540
2560 L=LW
2570 PP$=LEFT$(P$,L):P$=RIGHT$(P$,LEN(P$)-L)
2580 L$(LIN)=PP$
2590 LIN=LIN+1
2600 IFLEN(P$)>LWTHENL=LW:GOTO2540
2610 IFCRANDLEN(P$)>0THEN2520
2620 RETURN

```

Edit routine.

```

2630 IT=I:IFI>13THENV1=15:GOTO2670
2640 V1=I+2:PRINT@PP,;
2650 X=13:IFX>LN-1THENX=LN-1
2660 FORI=IT+1TOX:PRINTL$(I):NEXTI
2670 EL=V1-(IT>13)*(IT-13)-2
2680 Q$=" UP/DOWN ARROWS:MOVE ENTER:EDIT D,X:DELETE F:FIND C
LR:EXIT"
2690 PRINT@0,CHR$(30);Q$:PRINTSTRING$(63,"-");
2700 PRINT@V1*64-2,"<E";
2710 GOSUB160
2720 PRINTB$;B$;

```

Branch to proper sub-edit routine (cursor up, cursor down, delete, find or exit).

```

2730 IFC=10THEN2780ELSEIFC=CLTHENRETURNELSEIFC=68ANDV1<>3THEN283
0ELSEIFC=88ANDV1<>3THEN2880ELSEIFC=RTTHEN2960ELSEIFC=70THEN3240E
LSEIFC<>91THEN2700

```

Cursor up.

```

2740 IFV1>3THENV1=V1-1:EL=EL-1:GOTO2700

```



```

2750 IFEL=1THEN2700
2760 EL=EL-5:IFEL<1THENEL=1
2770 PRINT@127,"":FORJ=ELTOEL+12:PRINTL$(J):NEXTJ:GOTO2700
Cursor down.
2780 IFEL>=LN-1THEN2700
2790 EL=EL+1
2800 IFV1<15THENV1=V1+1:GOTO2700
2810 NN=4:IFNN>LN-EL-1THENN=LN-EL-1
2820 EL=EL+NN:PRINT@960,,:FORI=EL-NNTOEL:PRINTL$(I):NEXTI:GOTO26
90

```

Delete line.

```

2830 FORJ=ELTOLN-1:L$(J)=L$(J+1):NEXT:L$(LN)=""
2840 X=15-V1:IFX>LN-ELTHENX=LN-EL
2850 PRINT@V1#64-65,"":FORJ=ELTOEL+X:PRINTL$(J):NEXTJ
2860 IFEL=LN-1THENV1=V1-1:EL=EL-1
2870 LN=LN-1:GOTO2700

```

Delete to end of text.

```

2880 PRINT@0,CHR$(30);"DO YOU WANT TO DELETE FROM HERE TO THE EN
D OF THE TEXT? (Y/N) ";
2890 PRINT@V1#64-2,"<E";
2900 C$=INKEY$:IFC$=""THENPRINT@V1#64-2," ";GOTO2890
2910 IFC$<>"Y"ANDC$<>CHR$(121)THEN2690
2920 FORJ=ELTOLN:L$(J)="":NEXTJ:LN=EL:CH=1:SL=0
2930 PRINT@V1#64-65,"":FORJ=V1TO14:PRINT:NEXTJ
2940 PRINT@0,CHR$(30);@;
2950 PRINT@V1#64-66,"<E";GOTO2740
2960 L1=EL-4:IFL1<1THENL1=1
2970 L2=EL+4:IFL2>LNTHENL2=LN

```

Edit line.

```

2980 CLS:PRINT@0,CHR$(30);"TYPE NEW LINE BELOW (CLR-F TO FINISH)
":PRINTSTRING$(63,"-")
2990 FORJ=L1TOEL:PRINTL$(J):NEXTJ
3000 PRINT:PRINT:PRINT:PRINT
3010 FORJ=EL+1TOL2:PRINTL$(J):NEXTJ
3020 PRINT@{EL-L1+2}#64,;
3030 TLN=LN:LN=LN+1
3040 FORJ=1TO5:L$(TLN+J)="":NEXTJ
3050 C1=CH:S1=SL:CH=1:ED=-1
3060 GOSUB600
3070 CH=C1:SL=S1
3080 IFL$(LN)=""THENLN=LN-1
3090 NL=LN-TLN
3100 IFNL=1THEN3150
3110 IFNL>0THEN3140
3120 FORJ=ELDTOLN:L$(J)=L$(J+1):NEXTJ
3130 L$(LN)="":GOTO3220
3140 FORJ=LNTOL+1STEP-1:L$(J+NL-1)=L$(J):NEXTJ
3150 FORJ=0TOLN-1:L$(EL+J)=L$(LN+J):L$(LN+J)="":NEXTJ
3160 CX$=RIGHT$(L$(EL+NL-1),1)
3170 IFCX$=CR$THEN3220

```

MICROTEXT 2.0

```

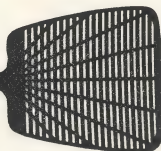
3180 IFCX$(<)" THENL$(EL+NL-1)=L$(EL+NL-1)+" "
3190 SS=1:L$=L$(EL+NL):LL=LEN(L$)
3200 IFMID$(L$,SS,1)<>" "ANDSS<=LLTHENSS=SS+1:GOTO3200
3210 IFLEN(L$(EL+NL-1))+SS<=LWTHENGOSUB2330:GOTO3230
3220 LN=TLN+NL-1
3230 RETURN
Search text.
3240 PRINT@0,CHR$(30);"SEARCH STRING:";S1$="":INPUTS1$
3250 IFS1$(<)" THENS$=S1$
3260 IFS$=""THEN2690
3270 IFPEEK(16396)=201THEN3310
3280 FORJ=EL-(S1$="")TOLN:IFINSTR(L$(J),S$)<>0THEN3330
3290 NEXTJ
3300 PRINT@0,CHR$(30);"NOT FOUND. HIT ANY KEY";GOSUB110:GOTO26
90
3310 FORJ=EL-(S1$="")TOLN:FORJ1=1TOLN(L$(J))-LEN(S$)+1:IFMID$(L
$(J),J1,LEN(S$))=S$THEN3330
3320 NEXTJ1,J:GOTO3300
3330 EL=J:V1=7:IFV1>EL+2THENV1=EL+2
3340 PRINT@127,"":FORJ=1TO12:PRINTL$(J+EL-V1+2):NEXTJ:GOTO2690
Error handling.
10000 PRINT@0,CHR$(30);"ERROR: CODE";ERR/2+1;" LINE";ERL;"-- P
RESS ANY KEY";
10010 IFINKEY$=""THEN10010
10020 P=P-1:RESUME550
Program exit.
11000 IFMD=3THENPOKE16916,0
11010 ONERRORGOTO0:END

```

SWAT

For TRS-80® MICROTEXT 2.0

TABLE



LINES	SWAT CODE	LENGTH	LINES	SWAT CODE	LENGTH
100 - 220	CW	341	2000 - 2110	GB	463
230 - 400	XD	285	2120 - 2230	WM	230
410 - 520	LN	251	2240 - 2350	BY	283
530 - 660	LQ	372	2360 - 2470	EL	273
670 - 1030	YN	362	2480 - 2590	JA	212
1040 - 1150	PA	216	2600 - 2710	JC	303
1160 - 1270	LX	273	2720 - 2830	EA	398
1280 - 1390	XF	258	2840 - 2950	RR	400
1400 - 1510	OK	278	2960 - 3070	CC	294
1520 - 1630	MJ	228	3080 - 3190	PX	319
1640 - 1750	TP	246	3200 - 3310	CZ	380
1760 - 1870	JD	195	3320 - 11010	DK	219
1880 - 1990	IW	227			

LIFE II



by Leo Christopherson
Documentation by Rich Bouchard

Life II is a game program for a 32K TRS-80® Model I or III. It is the bonus program on issue #42 SoftSide DV. See the coupon near the back of this booklet to order your disk.

John Conway presented his original game of "Life" in the October 1970 issue of Scientific American. *Life II* is a computerized version of the same game, with an additional multi-player game based on "Life".

You play "Life" on a rectangular grid of almost any size. You start with a given pattern of "cells" on this grid, and then apply the "rules of 'Life'" to each position on the grid, to perform the necessary action. Each iteration of this process is a "generation."

The rules of "Life" are simple:

- If a position is empty, and has exactly three neighbors, then a new cell is "born" at that location. A "neighbor" is a cell next to the the empty position or a cell being examined. An empty position or cell may have as many as eight neighbors.
- If a living cell has fewer than three neighbors, or more than four neighbors, it dies and disappears.
- All other empty locations remain empty, and all other cells remain "alive."

A "Life" example

First begin with a simple pattern:

	1	2	3	4	5
A					
B			●		
C			●		
D			●		
E					

We start with cells B3, C3, and D3 occupied. By applying the first rule of life to every empty cell, we find that new cells should be created at C2 and C4. By applying the second rule, we see that the cells at B3 and D3 both have only one neighbor, and will die. C3, with exactly two neighbors, will survive this generation. Thus we get:

	1	2	3	4	5
A					
B			—		
C		+	●	+	
D			—		
E					

	1	2	3	4	5
A					
B					
C		●	●	●	
D					
E					

A "+" represents a newly born cell, and a "—" represents a cell that has died this generation. At the start of the second generation, our grid will appear as shown at left.

Now the second generation can begin.

DV BONUS

LIFE II

DV BONUS

Life II includes two variations of the game of "Life." In one you can enter patterns of living cells, and then watch the computer run through generations of development. The second allows up to four players to compete in the "Battle of Life," where players place their androids on a small life grid, and allow them to battle for supremacy.

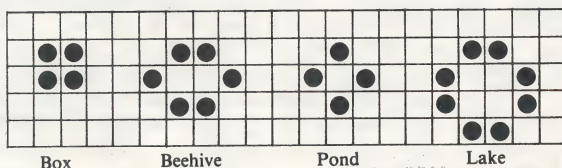
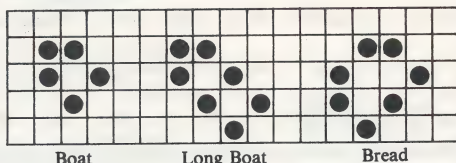
The Game of "Life"

The Game of Life provides a 62 by 42 unit grid for experiments with different life "organisms." It starts by presenting a blinking cursor on the screen. You can move this cursor across the screen with the arrow keys. Pressing "S" will place a cell at the current cursor position, and "R" will erase a cell. The "CLEAR" key will remove all living cells from the screen, and "Q" will restart the program.

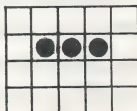
When you finish creating your life pattern, press "ENTER". The prompt will ask for the number of generations you wish to watch. Enter a number, and watch the show. While the show is going on, you may press "S" to change to the screen display.

After you have run a few patterns, you may notice some interesting results. You will probably find that certain specific cell patterns form on the screen. Some, called "still-lives," remain stationary. Others alternate between two or more different patterns; these are called "oscillators." Other patterns seem to grow forever, and yet others move across the screen in a specific pattern. Below are some common figures, and some of the more interesting ones:

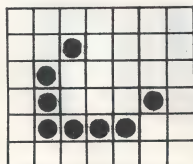
Still-lives:



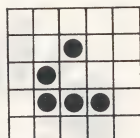
Other interesting patterns to try:



◀ "Blinker" was used as an example above, and will alternate between a vertical and horizontal line, each of three cells. It is the simplest oscillator.



▶ "Glider" will "glide" down and to the left, moving one unit every four generations.

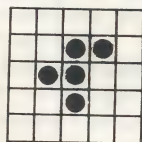


◀ "Spaceship" is a pattern of cells that travels to the left, moving one unit every four generations.

► This pattern will evolve into a pattern of four blinkers, called "traffic lights." A number of other interesting patterns consist of a single row of cells. For example, rows of 7, 8, 9 or 10 produce a complicated oscillator which repeats every fifteen generations.



► The famed "R-Pentomino." This deceptively simple pattern will grow to fill your entire video screen. If unlimited by the confines of a video screen, it would grow to occupy a huge area.



You can create many other interesting figures. For example, try a large "X" centered in the middle of the screen with each of the four lines drawn from the center being nine cells long. Large I-beam shapes, and various geometrical shapes (squares, rectangles, triangles, etc.) also produce interesting patterns.

The Battle of "Life"

In *The Battle of Life*, up to four players compete in a game based on the "Life" rules. A 10 x 5 grid is presented, with each square containing a number between 01 and 50. One of four life forms (uniped, biped, triped and quadruped) fills occupied cells. Players place life forms on the board by selecting the type of creature to be placed (1-4), followed by the two-digit location for it. The creature will then pop into place. This process of adding life forms will continue until you press "S" to start the action.

After you strike the "S" key, the computer examines each location on the grid, applying the normal rules of "Life" to each one. If a creature has too few or too many neighbors, it vanishes. If an empty location has exactly three neighbor organisms, a new one is "born." Since there are four different possible organisms on the screen, the computer must decide which type of organism the new creature will be. If the empty location had two or three neighbors of the same type, a corresponding creature type will appear. If its neighbors are of three different types, the new creature will be of the fourth type.

When the computer has run through five generations of "Life" the players will have another chance to add additional organisms. This process repeats until you type "Q".

The computer does not monitor how many pieces each player places, so the players must decide what rules they play by. One common game allows each player to place a specific number of creatures at the start, using a simple rotation. Every five turns each player adds one additional organism, if and only if at least one of the creatures remains alive. The process continues until either all creatures die off or only one type of creature remains.

For your first few games, we suggest allowing each player to place the following number of pieces at the start of each game:

Two-player game: 5

Three-player game: 4

Four-player game: 3

While playing this game, don't be surprised if you see some of the creatures on the screen move. In fact, the entire screen will appear to be teeming with life — the creatures move, look around, and speak to one another (attach an amplifier to your cassette AUX jack to hear this and other sound effects).

KING II



KING II

by John F. Gabriel

King II is a political war game for one or two players. A TRS-80 with 16K is necessary.

Two Player Description

Following your coronation as King, the Royal Advisors besiege you with all the problems facing the country. Each year numerous decisions affecting your ability to remain in power and the state of the kingdom confront you. To simplify the annual decision-making process, you have divided the turmoil into various phases: Land, Harvest, Riot, Natural Disasters, Economic and Political, War Preparation, and Battle. The order and importance of the phases varies from year to year; nevertheless you face the future with determination and boldness. As the years go by you learn from each decision and your country prospers....

The Land phase is now in progress. Since the kingdom has been acquiring land steadily, you can obtain no more. All available land has been settled, so you cannot sell any. Deciding to cultivate all of it ends the Land phase.

At Harvest time, the Royal Advisors inform you that not enough silos are available to store the unusually abundant harvest. Retaining enough to feed everyone, you sell off all the extra bushels for revenue. You will need this money for future exploits. The Harvest phase ends.

With full silos and prosperity throughout the land, the peasants are happy and well-fed. Since you have treated your people splendidly and made no erratic mistakes, no riots take place. But just in case they do, you sleep better at night knowing you have an ample number of Guardsmen to suppress a riot.

A stroke of luck! You were spared any natural disasters this year. Mother Nature treated you kindly and you will take full advantage of this.

"Life, Joy, Empire and Victory"

At last, the moment you've been waiting for — the Economic and Political phase. Inflation is at a modest level, the capitol city is spacious and beautiful, the Guard is more than ample — and you are bored. The time has come. You declare war on the hated barbarians in the neighboring kingdom.

In the War Preparation phase many important decisions must be made quickly: whether to enlist men, transfer Guardsmen to the Army, increase salaries, purchase arms factories, or purchase weapons. After a short period of reflection, you decide to buy the factories to reduce the price of weapons. The rest you will obtain later.

Difficulties arise as the Battle phase begins. Since you have few men and weapons, you cannot wreak havoc on the enemy. However, the war will go on and you will get your chance. The best strategy is a surprise attack on the opposing army. After a merciless raid on their camp, leaving them devastated and defenseless, you prepare to march to the capitol and claim the throne. But dangers await. The enemy may convince their allies to help retaliate. Worse yet, your people may respond to your warmongering and rise up in rebellion. Despite these dangers, you simply shrug it off, steadfast, bold, determined.

The Last Campaign

Many years have passed. The war has subsided and been forgotten. Reminiscing on past conquests, you feel stirring within you those old yearnings for the excitement and glory of years gone by. Perhaps the time has come to pass on the Crown to your son, who has now grown to young manhood. He too is anxious to prove himself on the field of battle. But you wish to die in honor, and with the Crown Prince's help, marshal the resources of the Empire for one last expedition of conquest. In the twilight of your reign, with your son at your side, you lead the armies into the final climactic battle. Fear of death does not concern you. In the end only the memory of your victories will live on.

Notes on Playing

King II was designed for two players. One person may play, but the challenge is greater with two people. The one-player version does not include the War Preparation phase nor the Battle phase, but the other phases will provide a challenge (try it and see).

When the game ends for a player, he receives a rating of his performance, based on the following goals:

Total Land	735,000 Acres
Salaries	\$110,000
Level of Discontent	0 = Lowest
Guards	880
Inflation	0 = Low
Fertile Land	420,000 Acres
City Land	231,000 Acres
Population	44,000
Grain	52,500,000 Bushels
Wealth	\$1,100,000,000

To get a perfect rating of 100%, you do not necessarily need to attain *all* of these goals. A good strategy is to achieve high scores in a few areas. In the two-player version, you should expect a lower rating, especially if you have fought many battles. When entering battle, troop morale is based on the following three goals:

Conscripts	3,000
Weapons	10,000
Salaries	50,000

You cannot go to battle without conscripts and weapons. Be careful of how many conscripts you employ and how much you pay them, because the annual payroll may plunge you into debt.

During play, variables interact in simple, meaningful ways. For example, buying factories decreases the cost of weapons. Reducing inflation makes the

price of new land less. However, grain sells at a lower price when you bring down inflation. Therefore, weigh the *effects* of your decisions carefully.

Lists of Variables

Main Variables

A-C, Z: Temporary
 F: Who goes first this year (1 = first player, 2 = second player)
 P: Number of Phases
 Q: Treasurer's Report Tab
 R: Secretary's Report Tab
 S: Land Phase Tab
 T: Turn
 U: Harvest Phase Tab
 W: State of War (0 = Peace, 1 = War)
 Y: Used for Dot Placement when aligning columns

A#-C#: Temporary
 PL: Players (1 = first player, 2 = second player, 3 = second player)
 X(): Phase Array
 A\$-C\$: Graphic Strings
 D\$,E\$: String Formats
 Z\$: Temporary Answer

#3 Variables

A3#: Acres Owned
 B3: Conscript and Guard Salary
 C3#: Conscripts
 D3: Discontent (0 = Content to 100 = Discontent)
 F3: Factories
 G3#: Guards
 H3: Death Status (0 = Alive, 1-7 = Dead)
 I3: Inflation
 K3#: Fertile Land
 L3#: City Land
 P3#: Population
 Q3#: Weapons
 R3#: Stored Grain
 S3: Silos
 W3#: Wealth
 Y3: Year
 Z3: Title
 C3\$: Country
 N3\$: Name of Player

Note: All of the variables above have two similarly named variables — one for each player. For example, A1# = Acres Owned by First Player and A2# = Acres Owned by Second Player.

Subroutine Driver

The key to understanding how the program is organized and works is the Subroutine Driver (lines 400-470). All the subroutines from line 500 on employ #3 variables. What this means is that, since the same process applies to both players, only one common subroutine is written. This single subroutine is called on, once for each game player. This calling process occurs in three steps:

1. The Subroutine Driver assigns the appropriate player's variables to the #3 variables.
2. The subroutine is called and performs its operations using only the #3 variables (with minor exceptions).
3. The Subroutine Driver assigns the #3 variables back to the same player's variables.



SS SS SS SS SS SS SS SS SS SS SS
 SS
 SS TRS-80 BASIC SS
 SS 'King II' SS
 SS Author: John F. Gabriel Jr. SS
 SS Copyright © 1983 SS
 SS SoftSide Publications, Inc SS
 SS
 SS SS SS SS SS SS SS SS SS SS SS



If you don't wish to type this program, it
 is available on issue #42 SoftSide CV
 and DV.

Initialization.

```
10 CLS:PRINTCHR$(23);TAB(12);"KING II":PRINT:PRINTTAB(5);"BY JOH
N F GABRIEL JR.":RANDOM:CLEAR200:A$=STRING$(64,42):B$=STRING$(2,
42)+CHR$(32):C$=CHR$(189)+CHR$(191)+CHR$(190):D$="###,###,###":E
$="$$$,$$$,$$$,$$$":Q=30:R=20:S=50:U=50
15 DEFINTY
20 PRINT:INPUT"NUMBER OF PLAYERS (1-2)";PL:IFPL<1ORPL>2THEN20ELS
ECLS:F=2:W=0:T=0:X(T)=9:GOSUB430:IFPL=2THENGOSUB460
30 PRINT:INPUT"WHAT IS THE NAME OF PLAYER 1";N1$:IFLEN(N1$)>10TH
ENGOSUB90:GOTO30ELSEIFPL=1THEN50
40 PRINT:INPUT"WHAT IS THE NAME OF PLAYER 2";N2$:IFLEN(N2$)>10TH
ENGOSUB90:GOTO40
50 PRINT"YOU HAVE ";:IFPL=2THENPRINT"BOTH ";
60 PRINT"BEEN CORONATED KING OF YOUR COUNTRY.":PRINT
70 PRINTN1$;:INPUT", GIVE YOUR COUNTRY A NAME";C1$:IFLEN(C1$)>10
THENGOSUB90:GOTO70ELSEIFPL=1THEN100
80 PRINTN2$;:INPUT", GIVE YOUR COUNTRY A NAME";C2$:IFLEN(C2$)>10
THENGOSUB90:GOTO80ELSE100
90 PRINT"KEEP NAMES LESS THAN 11 CHARACTERS LONG":RETURN
```

Set up phase order for one year; change turn.

```
100 T=0:X(T)=10:GOSUB400:F=F+1:IFF>2THENF=1
110 X(T)=8:GOSUB400:IFW=1THENP=7ELSEP=5
120 FORT=1TOP:X(T)=0:NEXT:FORT=1TOP:A=RND(P)
130 IFX(A)=0THENX(A)=T:NEXT:T=0:GOTO150ELSEA=A+1:IFA>PTHENA=1
140 GOTO130
```

Game cycle and game over.

```
150 T=T+1:IFT>PTHEN100ELSEGOSUB400:X(T)=11:GOSUB400:IFH1=0ANDH2=
0THEN150ELSEIFH1>0ANDH2>0THENCLS:PRINTCHR$(23);TAB(4);"LONG LIVE
THE KINGS!":PRINT:END
160 W=0:IFH1=0THEN180ELSEPRINT:INPUT"DO YOU WANT A NEW CROWN";Z$
:IFLEFT$(Z$,1)="Y"THENA=T:T=0:X(T)=9:GOSUB430:T=A:FORA=1TO16:PRI
NT:NEXT:PRINTCHR$(23);TAB(9);C$:PRINT:GOSUB910:GOTO150
170 IFPL=2THENPL=3:H1=0:CLS:PRINTCHR$(23);"BYE NOW.":PRINT:GOSUB
910:GOTO150ELSE2600
180 PRINT:INPUT"DO YOU WANT A NEW CROWN";Z$:IFLEFT$(Z$,1)="Y"THE
NA=T:T=0:X(T)=9:GOSUB460:T=A:FORA=1TO16:PRINT:NEXT:PRINTCHR$(23)
;TAB(9);C$:PRINT:GOSUB910:GOTO150
```

KING II

```
190 IFPL=2THENPL=1:H2=0:CLS:PRINTCHR$(23);"BYE NOW.":PRINT:GOSUB
910:GOTO150ELSE2600
```

Subroutine driver.

```
400 IFPL=1THEN430ELSEIFPL=3THEN460
410 IFF=1THENGOSUB430:GOSUB460ELSEGOSUB460:GOSUB430
420 RETURN
430 A3#=A1#:B3=B1:C3#=C1#:D3=D1:F3=F1:G3#=G1#:H3=H1:I3=I1:K3#=K1
#:L3#=L1#:P3#=P1#:Q3#=Q1#:R3#=R1#:S3=S1:W3#=W1#:Y3=Y1:Z3=Z1:C3#=
C1#:N3#=N1#:GOSUB450
```

```
440 A1#=A3#:B1=B3:C1#=C3#:D1=D3:F1=F3:G1#=G3#:H1=H3:I1=I3:K1#=K3
#:L1#=L3#:P1#=P3#:Q1#=Q3#:R1#=R3#:S1=S3:W1#=W3#:Y1=Y3:Z1=Z3:RETU
RN
```

```
450 DNX(T)GOTO1000,1100,1300,1400,1600,1800,2100,700,2400,2500,5
00
```

```
460 A3#=A2#:B3=B2:C3#=C2#:D3=D2:F3=F2:G3#=G2#:H3=H2:I3=I2:K3#=K2
#:L3#=L2#:P3#=P2#:Q3#=Q2#:R3#=R2#:S3=S2:W3#=W2#:Y3=Y2:Z3=Z2:C3#=
C2#:N3#=N2#:GOSUB450
```

```
470 A2#=A3#:B2=B3:C2#=C3#:D2=D3:F2=F3:G2#=G3#:H2=H3:I2=I3:K2#=K3
#:L2#=L3#:P2#=P3#:Q2#=Q3#:R2#=R3#:S2=S3:W2#=W3#:Y2=Y3:Z2=Z3:RETU
RN
```

Subroutines**End of game check.**

```
500 IFW3#<-600000000THENH3=1ELSEIFD3>RND(20)+70THENH3=2ELSEIFL3#
<P3#/6THENH3=3ELSEIFP3#<500THENH3=4ELSEIFI3>20+RND(10)THENH3=5EL
SEIFY3>30THENH3=6ELSEIFY3>RND(Y3)+24THENH3=7
```

```
510 IFH3=0THENRETURNELSECLS:GOSUB610:PRINTN3#;"", IN YEAR";Y3:PRI
NT:DNH3GOSUB2700,2800,2900,3000,3100,3200,3300:PRINT:GOTO3400
```

Title maker.

```
600 Z3=RND(12)
```

```
610 Z=Z3:IFZ=1THENPRINT"YOUR EXCELLENCY ";ELSEIFZ=2THENPRINT"YOU
R GREATNESS ";ELSEIFZ=3THENPRINT"YOUR HIGHNESS ";ELSEIFZ=4THENPR
INT"YOUR IMMENSENESS ";ELSEIFZ=5THENPRINT"YOUR IMPERIALNESS ";
```

```
620 IFZ=6THENPRINT"KING ";ELSEIFZ=7THENPRINT"YOUR LORDSHIP ";ELS
EIFZ=8THENPRINT"YOUR MAJESTY ";ELSEIFZ=9THENPRINT"YOUR MIGHTYNES
S ";ELSEIFZ=10THENPRINT"YOUR ROYALNESS ";ELSEIFZ=11THENPRINT"RUL
ER ";ELSEIFZ=12THENPRINT"SIRE ";
```

```
630 RETURN
```

Treasurer's report.

```
700 CLS:PRINTA#;:PRINT@63,B#;:GOSUB600:PRINTN3#;"", I YOUR HUMBLE
TREASURER, DO BEG":PRINT@127,B#;"TO REPORT TO YOU, THE STATUS O
F ";C3#;" IN YEAR";Y3;":":PRINT@191,A#;B#;"POPULATION";:PRINTTAB
(Q);USING@5;P3#:Y=268:GOSUB800:PRINT@319,B#;"CASH RESERVES";
```

```
710 PRINTTAB(Q-8);USING@5;W3#:Y=335:GOSUB800:PRINT@383,B#;"TOTAL
LAND (IN ACRES)";:PRINTTAB(Q);USING@5;A3#:Y=407:GOSUB800:PRINT
@447,B#;"CITY PROPERTY (IN ACRES)";:PRINTTAB(Q);USING@5;L3#:Y=47
4:GOSUB800:PRINT@511,B#;"FULL SILOS";
```

```
715 PRINTTAB(Q);USING@5;INT(R3#/10000):Y=524:GOSUB800
```

```
720 PRINT@575,B#;"NATIONAL GUARDS";:PRINTTAB(Q);USING@5;G3#:Y=59
3:GOSUB800:PRINT@639,B#;"INFLATION (%)";
```



```

725 IF I3<10 THEN PRINT TAB(Q*9); I3 ELSE PRINT TAB(Q*8); I3
730 Y=655:GOSUB 800:PRINT@703,B$;"THE CITIZENS ARE ";:IFD3<10 THE
TTAB(S);USINGD$;A#:Y=681:GOSUB 800:PRINT@704,"HOW MANY ACRES OF S
EED DO YOU WANT TO PLANT":PRINT@747,;:B#=A#:INPUTB$;:IFB#<0 OR B#>A
# THEN 1080 ELSE K3#=B#:RETURN
1100 CLS:GOSUB 610:PRINTN3$;," THIS IS THE HARVEST PHASE.":A#=100
000+INT(I3*1000):B=20+INT(I3*.2):IFW3#<0 THEN 1130 ELSE B#=W3#/A#:I
FB#>30000 THEN A=30000 ELSE A=INT(B#)
NPRINT"HAPPY";ELSE IFD3<30 THEN PRINT"CONCERNED";ELSE IFD3<50 THEN PRI
NT"WORRIED";ELSE IFD3<70 THEN PRINT"DISCONTENT";ELSE PRINT"OUTRAGED"
;
740 PRINT"" WITH YOUR PERFORMANCE";:PRINT@767,B$;:PRINT@768,A$;:
GOSUB 910:IFPL<>2 THEN RETURN

```

Secretary's report.

```

750 CLS:PRINTA$;:PRINT@63,B$;:GOSUB 610:PRINTN3$;," I YOUR SECRET
ARY OF WAR,":PRINT@127,B$;"DO BEG TO REPORT THE WAR STATUS:":PRI
NT@191,A$;B$;"CONSCRIPTS";:PRINTTAB(R);USINGD$;C3#;Y=268:GOSUB 80
0:PRINT@319,B$;"FACTORIES";:PRINTTAB(R);USINGD$;F3;Y=331:GOSUB 80
0
760 PRINT@383,B$;"WEAPONS";:PRINTTAB(R);USINGD$;Q3#;Y=393:GOSUB 8
00:PRINT@447,B$;"WE ARE AT ";:IFW=0 THEN PRINT"PEACE".ELSE PRINT"
WAR'."
770 PRINT@511,B$;:PRINT@512,A$;:GOTO 910

```

Make a row of dots.

```

800 Y=Y+15360
810 IF PEEK(Y)=46 THEN Y=Y+1:GOTO 810 ELSE IF PEEK(Y)=32 THEN POKEY,46:Y=
Y+1:GOTO 810 ELSE RETURN

```

Phase end response.

```

900 PRINT"YOU DON'T HAVE THE RESOURCES TO CONTINUE THIS PHASE."
910 INPUT"PRESS <ENTER> TO CONTINUE";Z$;RETURN

```

Buy land.

```

1000 CLS:GOSUB 610:PRINTN3$;," THIS IS THE LAND PHASE.":A=20+INT(
I3*.2):PRINT:PRINT"LAND IS SELLING AT (PER ACRE)";:PRINTTAB(S-B)
;USINGE$;A;:Y=157:GOSUB 800:IFA3#>=800000 THEN PRINT@192,"YOU CANN
O ACQUIRE ANY MORE LAND THROUGH PURCHASES.":GOTO 1050
1010 IFW3#<0 THEN PRINT@192,"YOU DON'T HAVE ENOUGH MONEY TO ACQUI
RE LAND":GOTO 1050 ELSE A#=W3#/A:IFA#>A3#>1000000 THEN A#=1000000-A3#
1020 IFA#>32000 THEN B=32000 ELSE B=INT(A#)
1030 PRINT@256,"THE MOST LAND YOU MAY BUY (IN ACRES) IS";:PRINTT
AB(S);USINGD$;B;Y=295:GOSUB 800:PRINT@320,"HOW MANY ACRES DO YOU
WANT TO BUY":PRINT@353,;:C=B:INPUTC:IFC<0 OR C>B THEN 1030 ELSE IFC=0
THEN 1050
1040 W3#=W3#-C*A:A3#=A3#+C:GOTO 1070

```

Sell unused land.

```

1050 A#=A3#-L3#-K3#:IFA#<0 THEN 1070 ELSE IFA#>32000 THEN B=32000 ELSE
B=INT(A#)
1060 PRINT@448,"THE MOST LAND YOU MAY SELL (IN ACRES) IS";:PRINT
TAB(S);USINGD$;B;Y=488:GOSUB 800:PRINT@512,"HOW MANY ACRES DO YOU
WANT TO SELL":PRINT@546,;:C=0:INPUTC:IFC<0 OR C>B THEN 1060 ELSE W3#
=W3#+C*A:A3#=A3#-C

```

KING II

Plant seed.

1070 A#=A3#-L3#: IFA#<=0 THEN PRINT@640, "YOU HAVE NO LAND AVAILABLE FOR PLANTING SEED.": GOTO 910

1080 PRINT@640, "THE MOST LAND YOU MAY PLANT (IN ACRES) IS": PRINT

Buy silos.

1110 IFA#<=0 THEN 1130 ELSE PRINT@128, "THE NUMBER OF SILOS YOU POSSESS ARE": PRINT TAB(U); USING D\$; S3: Y=163: GOSUB 800: PRINT@192, "EACH SILO CAN HOLD 10,000 BUSHELS OF GRAIN.": PRINT "EACH SILO COST": PRINT TAB(U-8); USING E\$; A#: Y=270: GOSUB 800

1120 PRINT@320, "YOU CAN AFFORD TO BUILD": PRINT TAB(U); USING D\$; A: Y=343: GOSUB 800: PRINT@384, "HOW MANY SILOS DO YOU WANT TO BUILD": PRINT@419, : C=0: INPUT C: IFC<0 OR C>A THEN 1120 ELSE W3#=W3#-A#*C: S3=S3+C

Harvest.

1130 A=RND(5): PRINT@512, "YOU HAD ": IFT=1 OR T=2 THEN A=A+15: PRINT "A N EARLY": ELSE IFT=3 OR T=4 OR T=5 THEN A=A+20: PRINT "A NORMAL": ELSE A=A+10: PRINT "A LATE":

1140 PRINT "HARVEST, YIELDING": A; "BUSHELS PER ACRE": A#=K3#*A: PRINT "FOR A TOTAL HARVEST (IN BUSHELS) OF": PRINT TAB(U); USING D\$; A#: Y=611: GOSUB 800: A=Y+30+RND(10): B#=P3#*A

1150 PRINT@640, "TO FEED YOUR HUNGRY CITIZENS, YOU NEED": PRINT TAB(U); USING D\$; B#: Y=678: GOSUB 800: IFA#>B# THEN 1200 ELSE IFR3#<=0 THEN 1180

1160 A#=A#+R3#: R3#=0: D3=D3+1: IFA#>B# THEN R3#=A#-B#: A#=B#

1170 PRINT@704, "EXTRA BUSHELS HAVE BEEN SEIZED FROM THE SILOS": PRINT "GIVING YOU A TOTAL YIELD (IN BUSHELS) OF": PRINT TAB(U); USING D\$; A#: Y=808: GOSUB 800: IFA#>B# THEN 1200

Buy grain.

1180 PRINT@832, "HOW MANY BUSHELS OF GRAIN WOULD YOU LIKE @": PRINT TAB(U-8); USING E\$; B: Y=874: GOSUB 800: PRINT@896, : C#=0: INPUT "PER BUSHEL": C#: IFC#<0 OR C#>100000000 THEN 1150 ELSE W3#=W3#-C#*B: A#=A#+C#: IFA#>B# THEN 1200

1190 C#=P3#-INT(A#/A): P3#=P3#-C#: D3=D3+C#/10+1: PRINT@960, "YOU HAVE STARVED TO DEATH": USING D\$; C#: PRINT "OF YOUR PEOPLE": PRINT: GOTO 910

Feed people, sell and store grain.

1200 A#=A#-B#

1210 PRINT@960, "YOUR PEOPLE ARE FULL AND CONTENT. ": GOSUB 910: IFA#<=0 THEN RETURN ELSE SCL3: GOSUB 610: PRINT N3#, "": PRINT@128, "YOU HAVE A GRAIN SURPLUS (IN BUSHELS) OF": PRINT TAB(U); USING D\$; A#: Y=168: GOSUB 800

1215 PRINT@192, "UNSOLD BUSHELS WILL BE STORED IN ANY AVAILABLE SILOS."

1220 PRINT@320, "HOW MANY BUSHELS DO YOU WANT TO SELL @": PRINT TAB(U-8); USING E\$; B: Y=358: GOSUB 800: PRINT@384, : C#=A#: INPUT "PER BUSHEL": C#: IFC#<0 OR C#>A# THEN 1220

1230 W3#=W3#+C#*B: A#=A#-C#: R3#=R3#+A#: C#=S3+10000: IFR3#>C# THEN PRINT "YOU HAD TO DUMP": USING D\$; R3#-C#: PRINT "BUSHELS.": R3#=C#: GOTO 910 ELSE RETURN

Earthquake.

```
1300 IFD3<10THENRETURNELSEA=INT(D3/5):CLS:PRINTCHR$(23);:FORB=1T
0A:PRINTTAB(12);"RIOT":FORZZ=1T050:NEXTZZ:NEXT:B=63##.009:PRINT:
60SUB610:PRINTN3#;",";IFA<BTHEN1310ELSE1330
1310 PRINT"THE RIOT HAS BEEN CONTAINED BY":PRINT"THE NATIONAL GU
ARD.":D3=D3-ABS(A-B):IFD3<0THEND3=0
1320 PRINT:GOTO910
1330 PRINT"THE RIOT IS OUT OF CONTROL!":D3=D3+RND(A):GOTO1320
1400 A=RND(100):IFA<80THENRETURNELSECLS:60SUB610:PRINTN3#;",";DIS
ASTER HAS STRUCK.":PRINT:A=RND(3):0NAGOTO1410,1470,1500
1410 IFP3#>5000THENA=5000ELSEA=INT(P3#)
1420 A=RND(RND(RND(RND(RND(A))))):P3#=P3#-A:PRINT"AN EARTHQUAKE
HAS KILLED";USINGD$;A;:PRINT" OF YOUR PEOPLE.":A=RND(RND(RND(200
0))):L3#=L3#-A:IFL3#<=0THENL3#=0
1430 PRINTA;"ACRES OF CITY PROPERTY WERE DESTROYED.":A=RND(100):
IFA>250R3#<=0THEN1450ELSEA=RND(RND(RND(RND(S3)))):S3=S3-A:R3#=R3
#-10000#A:IFR3#<0THENR3#=0
1440 PRINTA;"OF YOUR SILOS WERE DECIMATED, ALONG WITH THE":PRINT
"GRAIN STORED IN THEM."
1450 IFC3#>P3#THENC3#=P3#
1460 IF63#>P3#THENG3#=P3#
1465 PRINT:GOTO910
```

Tornado or hurricane.

```
1470 A=RND(100):P3#=P3#-A:B=RND(2):IFB=1THENPRINT"A TORNADO HAS
RIPPED THROUGH ";C3#;",";ELSEPRINT"A HURRICANE HAS BLOWN ASHORE,"
1480 PRINT"KILLING";A;:PRINT" OF YOUR PEOPLE.":IFK3#>30000THENA=30000EL
SEA=INT(K3#)
1490 A=RND(RND(RND(A))):K3#=K3#-A:PRINT"YOU HAVE LOST";USINGD$;A
;:PRINT" ACRES OF SEED.":GOTO1450
```

Flood.

```
1500 IFK3#>30000THENA=30000ELSEA=INT(K3#)
1510 A=RND(RND(A)):(K3#=K3#-A:PRINT"A FLOOD HAS SWAMPED";USINGD$;
A;:PRINT" ACRES OF FERTILE LAND.":PRINT:GOTO910
```

Fight inflation.

```
1600 CLS:60SUB610:PRINTN3#;",";PRINT"THIS IS THE ECONOMIC & POLI
TICAL PHASE.":PRINT:PRINT"YOUR OPTIONS ARE:":PRINT:PRINT"1 - DO
NOTHING":PRINT"2 - FIGHT INFLATION":PRINT"3 - EXPAND CITY":PRINT
"4 - INCREASE NATIONAL GUARD":IFPL=2ANDW=0THENPRINT"5 - DECLARE
WAR"
1610 IFPL=2ANDW=1THENPRINT"6 - ASK FOR A TRUCE"
1620 PRINT:A=1:INPUT"YOUR CHOICE PLEASE";A:IFA<10RA>6THEN1620ELS
EIFA=5AND(PL<>2ORW=1)THEN1620ELSEIFA=6AND(PL<>2ORW=0)THEN1620
1630 0NAGOTO1690,1640,1700,1730,1770,1780
1640 PRINT:IFI3<=0THENPRINT"INFLATION IS UNDER CONTROL.":GOTO910
ELSEIFW3#<=15000THEN900ELSEPRINT"INFLATION IS AT";I3;"% AND IS U
NPREDICTABLE TO COMBAT.":A#=I3#10000:IFA#>W3#THENA#=#W3#
1650 PRINT"YOU MAY INVEST UP TO";USINGE$;A#;:PRINT" TO FIGHT IT.
":B#=#A#:INPUT"HOW MUCH (IN $) DO YOU WANT TO INVEST";B#:IFB#<0OR
B#>A#THEN1650ELSEA=B#/10000:IFA>300THENA=300
1660 A=RND(A):IFA>100THENA=100
```

KING II

```
1670 W3#=W3#-B#;I3=I3-A;IFI3<-5THENI3=-5
```

```
1680 D3=D3-A;IFD3<0THEND3=0
```

```
1690 RETURN
```

Expand city.

```
1700 A#=A3#-K3#-L3#;IFA#>32000THENA=32000ELSEA=INT(A#)
```

```
1710 PRINT:IFA<0THEN900ELSEPRINT"YOU HAVE:":PRINTUSINGD$;L3#;;P  
RINT" ACRES OF CITY LAND OUT OF";USINGD$;A3#;;PRINT" ACRES OWNED  
.";PRINT"YOU MAY SET ASIDE";USINGD$;A#;;PRINT" ADDITIONAL ACRES."
```

```
1720 PRINT:PRINT"HOW MANY ADDITIONAL ACRES DO YOU WANT";PRINT"TO  
ALOT ";C3#;;B=A:INPUTB:IFB<0ORB>ATHEN1720ELSEL3#=L3#+B:RETURN
```

```
1730 PRINT:A#-P3#-C3#-B3#;IFA#<=00RW3#<B3THEN900ELSEB#-W3#/B3:IF  
A#>B#THENA#-B#
```

Increase guard.

```
1740 IFA#>1000THENA=1000ELSEA=INT(A#)
```

```
1750 IFA<0THENPRINT"YOU HAVE ENOUGH GUARDS FOR NOW.":GOTO900ELS  
EPRINT"YOU HAVE:":PRINTUSINGD$;G3#;;PRINT" NATIONAL GUARDS AND P  
AY EACH";USINGE$;B3:PRINT"YOU MAY OBTAIN";USINGD$;A#;;PRINT" MORE  
."
```

```
1760 B=A:INPUT"HOW MANY MORE GUARDS DO YOU WANT";B:IFB<0ORB>ATHE  
N1760ELSEG3#=G3#+B:D3=D3+2:RETURN
```

Declare war.

```
1770 W=1:D3=D3+2:I3=I3-2:PRINT:PRINT"THIS MEANS WAR!":P=P+2:X(P-  
1)=6:X(P)=7:GOTO910
```

Truce.

```
1780 PRINT:GOSUB610:IFN3$=N1$THENPRINTN2$;",";ELSEPRINTN1$;","
```

```
1790 INPUT"DO YOU ACCEPT YOUR NEIGHBOR'S TRUCE";Z$:IFLEFT$(Z$,1)  
="Y"THENW=0:D3=D3-1:PRINT"A CELEBRATION IS TAKING PLACE.":PRINTE  
LSEPRINT"THE WAR TRUDGES ON...":PRINT:IFN3$=N1$THEND2=D2+3ELSEDI  
=D1+3
```

```
1795 GOTO910
```

```
1800 CLS:GOSUB610:PRINTN3$;",";PRINT"THIS IS THE WAR PREPARATION  
PHASE.":PRINT:PRINT"YOUR ALTERNATIVES ARE:":PRINT:PRINT"1 - DO
```

Enlist soldiers.

```
NOTHING":PRINT"2 - ENLIST SOLDIERS":PRINT"3 - TRANSFER GUARDS TO  
ARMY":PRINT"4 - INCREASE SALARIES":PRINT"5 - PURCHASE FACTORIES  
"
```

```
1805 PRINT"6 - PURCHASE WEAPONS":PRINT
```

```
1810 PRINT:A=1:INPUT"YOUR CHOICE, PLEASE";A:IFA<10RA>6THEN1810EL  
SEPRINT:ONAGOTO1690,1820,1880,1930,1950,1990
```

```
1820 A#-P3#-B3#-C3#;IFA#<00RW3#<B3THEN900ELSEB#-W3#/B3:IFA#>B#TH  
ENA#-B#
```

```
1830 IFA#+C3#>P#/2THENA#-P3#/2-C3#
```

```
1840 IFA#+C3#>500*Y3THENA#-500*Y3-C3#
```

```
1850 IFA#>5000THENA=5000ELSEA=INT(A#)
```

```
1860 IFA<0THENPRINT"YOU HAVE YOUR LIMIT OF SOLDIERS.":GOTO910EL  
SEPRINT"YOU HAVE:":USINGD$;C3#;;PRINT" SOLDIERS AND PAY EACH";US  
INGE$;B3:PRINT"YOU MAY OBTAIN";USINGD$;A#;;PRINT" MORE.":PRINT
```

```
1870 B=A:INPUT"HOW MANY MORE SOLDIERS DO YOU WANT";B:IFB<0ORB>AT  
HEN1870ELSEC3#=C3#+B:RETURN
```


Transfer guards.

```
1880 IFB3#<=0THEN900ELSEA#=#B3#:#IFA#>C3#>P3#/#3THENA#=#P3#/#3-C3#
1890 IFA#>C3#>500#Y3THENA#=#500#Y3-C3#
1900 IFA#>5000THENA=#5000ELSEA=INT(A#)
1910 IFA#<=0THENPRINT"YOU HAVE YOUR FILL OF TRANSFERS.":GOTO910EL
SEPRINT"YOU HAVE";USINGD#;A#;:PRINT" GUARDS THAT YOU CAN TRANSFER
."
1920 B=A:INPUT"HOW MANY GUARDS DO YOU WANT TO TRANSFER";B:IFB<0O
RB>ATHEN1920ELSEC3#=#C3#:#B:#B3#=#B3#-B:RETURN
```

Buy factories.

```
1930 PRINT"YOU PAY YOUR MEN ";USINGE#;B3:PRINT"AND MAY NOT PA
Y THEM MORE THAN $500,000":PRINT
1940 A=0:INPUT"BY HOW MUCH WOULD YOU LIKE TO INCREASE THEIR PAY"
;A:#IFA<0ORA>A+B3>500000THEN1940ELSEB3=#B3+A:RETURN
1950 A#=#1000000+I3#*100000:B#=#W3#/#A#:#IFB#<=0THEN900ELSEIFB#>30000
THENA=#30000ELSEA=INT(B#)
1960 IFF3+A>500000THENA=#500000-F3
1970 IFA#<=0THENPRINT"YOU HAVE ENOUGH FACTORIES.":GOTO910ELSEPRIN
T"YOU HAVE";USINGD#;F3;:PRINT" FACTORIES AND CAN BUY";USINGD#;A#;
:PRINT" MORE":PRINT"AT A PRICE OF";USINGE#;A#;:PRINT" A PIECE.":P
RINT
1980 B=A:INPUT"HOW MANY FACTORIES DO YOU WANT";B:IFB<0ORB>ATHEN1
980ELSEW3#=#W3#-A#:#B:#F3=#F3+B:RETURN
```

Buy weapons.

```
1990 A#=#2000000+I3#*200000:IFF3>0THENA#=#A#/#F3
2000 B#=#W3#/#A#:#IFB#>30000THENA=#30000ELSEA=INT(B#)
2010 IFA#<=0THEN900ELSEPRINT"YOU HAVE";USINGD#;Q3#;:PRINT" WEAPON
S.":PRINT"NEW WEAPONS COST";USINGE#;A#;:PRINT" A PIECE.":PRINT"TH
E MOST YOU MAY BUY IS";USINGD#;A#;:PRINT
2020 B=A:INPUT"HOW MANY NEW WEAPONS DO YOU WANT";B:IFB<0ORB>ATHE
N2020ELSEW3#=#W3#-A#:#B:#Q3#=#Q3#+B:RETURN
```

Destroy men and weapons.

```
2100 IFC3#<=0ORQ3#<=0THENRETURNELSECLS:GOSUB610:PRINTN3#;," THIS
IS THE BATTLE PHASE.":PRINT:B=#Q3#/#300:#IFB>33THENB=33
2110 A=B:B=#C3#/#90:#IFB>33THENB=33
2120 A=A+B:B=#B3/#1500:#IFB>33THENB=33
2130 A=A+B:A=INT(A):PRINT"MORALE IS ";:#IFA<20THENPRINT"VERY LOW.
"ELSEIFA<40THENPRINT"LOW."ELSEIFA<60THENPRINT"NORMAL."ELSEIFA<80
THENPRINT"HIGH."ELSEPRINT"VERY HIGH."
2140 PRINT:PRINT"YOUR CHOICES ARE:":PRINT:PRINT"1 - DESTROY MEN
AND WEAPONS":PRINT"2 - BOMB FACTORIES":PRINT"3 - BOMB SILOS":PRI
NT"4 - BOMB CAPITOL":PRINT
2150 B=0:INPUT"YOUR ORDERS, CHIEF COMMANDER";B:IFB<1ORB>4THEN215
0ELSEPRINT:IFN3#=#N1#THENZ=1ELSEZ=2
2160 DNB#GOTO2170,2220,2250,2290
2170 B=#9#A+RND(A):IFZ=1ANDB>C2#THENB=C2#ELSEIFZ=2ANDB>C1#THENB=C
1#
2180 IFZ=1THENP2#=#P2#-B:C2#=#C2#-B:D2=#D2+B/100ELSEP1#=#P1#-B:C1#=#C
```

KING II

```

10-B:D1=D1+B/100
2190 PRINT"YOU HAVE INFLICTED";USINGD$;B;:PRINT" CASUALTIES TO T
HE ENEMY":B=B+RND(B+1):IFZ=1ANDB>Q2#THENB=Q2#ELSEIFZ=2ANDB>Q1#TH
ENB=Q1#
2200 IFZ=1THENQ2#=Q2#-BELSEQ1#=Q1#-B
2210 PRINT"AND HAVE DESTROYED";USINGD$;B;:PRINT" OF THEIR WEAPON
S.":PRINT:GOTO910

```

Bomb factories.

```

2220 B=RND(RND(A))*7:IFZ=1ANDB>F2THENB=F2ELSEIFZ=2ANDB>F1THENB=F
1
2230 IFZ=1THENF2=F2-B:D2=D2+B/50ELSEF1=F1-B:D1=D1+B/50
2240 PRINT"YOU KNOCKED OUT";B;"ENEMY FACTORIES.":GOTO910

```

Bomb silos.

```

2250 B=RND(A)*20:IFZ=1ANDB>S2THENB=S2ELSEIFZ=2ANDB>S1THENB=S1
2260 IFZ=1THENS2=S2-B:D2=D2+B/50ELSE1=S1-B:D1=D1+B/50
2270 PRINT"YOU PICKED OFF";USINGD$;B;:PRINT" ENEMY SILOS.":IFZ=1
ANDR2#>10000*S2THENR2#=10000*S2ELSEIFZ=2ANDR1#>10000*S1THENR1#=1
0000*S1
2280 GOTO910

```

Bomb capitol.

```

2290 B=RND(A)*100:IFZ=1ANDB>L2#THENB=L2#ELSEIFZ=2ANDB>L1#THENB=L
1#
2300 IFZ=1THENL2#=L2#-B:D2=D2+B/500:C=RND(B/10):IFC>P2#THENC=P2#
2310 IFZ=2THENL1#=L1#-B:D1=D1+B/500:C=RND(B/10):IFC>P1#THENC=P1#
2320 IFZ=1ANDB2#>P2#THENG2#=P2#ELSEIFZ=2ANDB1#>P1#THENG1#=P1#
2330 PRINT"YOU REDUCED TO RUBBLE";USINGD$;B;:PRINT" ACRES OF THE
ENEMY'S CAPITOL":PRINT"KILLING";USINGD$;C;:PRINT" CIVILIANS.":P
RINT:GOTO910

```

Initial setting of variables.

```

2400 A3#=1000:B3=500:C3#=0:F3=0:G3#=0:H3=0:I3=0:K3#=500:L3#=300:
P3#=1000:Q3#=0:R3#=100000:S3=10:W3#=30000:Y3=0:IFPL=2THEND3=9ELS
ED3=21+RND(5)
2410 RETURN

```

Yearly update.

```

2500 Y3=Y3+1:I3=I3+RND(3):A#=I3/100*B3:IFA#>30000THENA=30000ELSE
A=INT(A#)
2505 B3=B3+A:D3=D3-1:IFD3<0THEND3=0
2510 A#=P3#/10:IFA#>30000THENA=30000ELSEA=INT(A#)
2520 A=A+RND(A):P3#=P3#+A:W3#=W3#+L3#*15-C3#*B3-B3#*B3:RETURN

```

Game end messages.

```

2600 CLS:PRINTCHR$(23);TAB(5);"LONG LIVE THE KING!":PRINT:END
2700 PRINT"THE LOANSHARKS SLIT YOUR THROAT.":PRINT"YOU SHOULD HA
VE STAYED OUT OF DEBT.":RETURN
2800 PRINT"YOUR PEOPLE ARE FED UP WITH YOUR RULE AND":A=RND(3):I
FA=1THENPRINT"HAVE HANGED YOU."ELSEIFA=2THENPRINT"HAVE SHOT YOU
IN FRONT OF A FIRING SQUAD."ELSEPRINT"HAVE BURIED YOU ALIVE."
2810 RETURN
2900 PRINT"DUE TO THE LACK OF LIVING SPACE, YOU'VE BEEN SQUEEZED
TO DEATH.":RETURN

```



```

3000 PRINT"TOO MANY PEOPLE DIED UNDER YOUR RULE.":GOTO2800
3100 PRINT"INFLATION SKYROCKETED YOU OUT OF OFFICE.":PRINT"YOU A
RE LOST IN SPACE...":RETURN
3200 PRINT"YOU RETIRED FROM OFFICE AFTER A LONG USEFUL LIFE AS K
ING.":PRINT"YOUR PEOPLE LOVE AND RESPECT YOU SO MUCH THAT YOUR P
ICTURE":PRINT"WILL APPEAR ON CURRENCY IN ";C3$;".":PRINT:PRINT"Y
OU LIVED OUT THE REST OF YOUR LIFE IN SPLENDID RETIREMENT.":RETU
RN
3300 PRINT"YOU DIED OF NATURAL CAUSES IN OFFICE.":PRINT"YOU WILL
ALWAYS BE REMEMBERED BY YOUR PEOPLE.":PRINT:PRINT"MAY GOD REST
YOUR SOUL...":RETURN

```

Rating.

```

3400 B#=A3#/700:IFB#>1050THENB#=1050
3410 A=B#:B=B3/2000:IFB#>55THENB=55
3420 A=A+B:B=(100-D3)*10:IFB<0THENB=0ELSEIFB>1050THENB=1050
3430 A=A+B:B=B3#/8:IFB#>110THENB#=110
3440 A=A+B#:B=100-13:IFB>110THENB=110
3450 A=A+B:B=K3#/800:IFB#>525THENB#=525
3460 A=A+B#:B=L3#/700:IFB#>330THENB#=330
3470 A=A+B#:B=P3#/40:IFB#>1100THENB#=1100
3480 A=A+B#:B=R3#/50000:IFB#>1050THENB#=1050
3490 A=A+B#:B=W3#/1000000:IFB#>1100THENB#=1100
3500 A=A+B#:A=INT((A*1000)/6050):A=A/10:IFA>100THENA=100
3510 PRINT"YOUR RATING IS";A;"%":IFA=100THENPRINT:PRINT"CONGRATU
LATIONS ON A PERFECT SCORE."
3520 PRINT:GOTO910

```



SWAT TABLE

For TRS-80® KING II

LINES	SWAT CODE	LENGTH
10 - 70	AE	562
80 - 160	MC	503
170 - 440	RT	559
450 - 510	UZ	550
600 - 700	YY	527
710 - 740	DW	532
750 - 1000	YW	689
1010 - 1060	KH	561
1070 - 1110	FF	569
1120 - 1150	LT	508
1160 - 1210	HN	639
1215 - 1310	RZ	503
1320 - 1440	EP	516

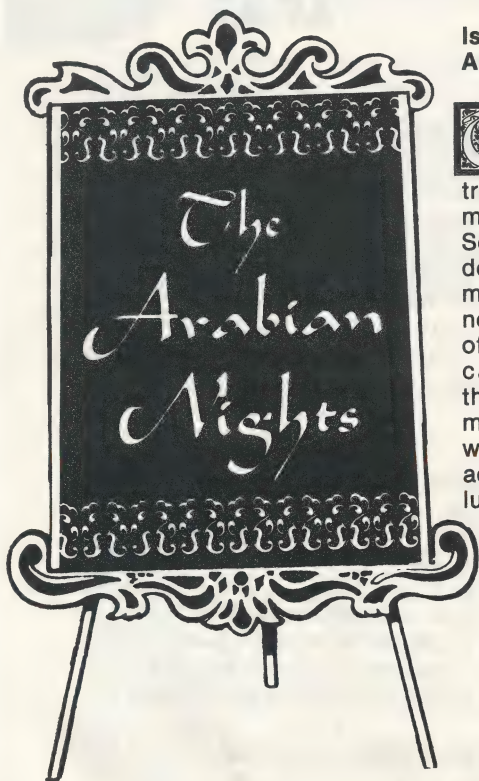
LINES	SWAT CODE	LENGTH
1450 - 1600	UZ	632
1610 - 1670	FM	525
1680 - 1750	VI	561
1760 - 1800	TX	573
1805 - 1880	JA	537
1890 - 1950	NF	508
1960 - 2020	NL	549
2100 - 2170	LX	537
2180 - 2260	LS	510
2270 - 2400	RR	565
2410 - 2900	VU	557
3000 - 3400	RZ	521
3410 - 3520	OB	478

KING II

SoftSideTM ADVENTURE SERIES

Issue 42 Adventure: Arabian Nights

The daughter of the Caliph of Baghdad lies in a death-like trance, the victim of the evil magician Roxor's spell. Scattered throughout the desert are the tools and magical items you will need to acquire the eggshell of the great Roc, which can restore health to the Princess. Beware of monsters and poisoned water, and heed well the advice of the seer! Good luck, sahib!



SoftSide Adventure Series **CV** **DV**

What would you say to a program that asks, "What do you want to do?" Well, you might say, "GET RUBY" or "KILL GIANT", because that's how the *SoftSide Adventure Series* works.

Each issue, the latest Adventure takes you to another world of fantasy, puzzles, and thrills. Your first task is often simple survival — and even that basic feat can be daunting until you figure out the *right* way to do it. You'll have to be ingenious and persevering, and your rewards will be great.

To "win" a fantasy/adventure game, you must solve the author's devious puzzles, and overcome the obstacles that confront you — whether they be dragons or desperadoes. Death, should it come, is transitory — just re-run the program to live again!

Experienced adventurers create detailed maps of each world as they search for solutions, but you can omit this exercise if your memory is exceptional. Express your wishes with one- or two-word commands, like "LOOK", "DROP KNIFE", or "GET RUBY". Use "I" to get an inventory of your possessions. This issue's Adventure, *Arabian Nights*, features the commands "SAVE GAME" and "LOAD GAME". These permit you to try to solve the Adventure in more than one sitting. You'll also want to save the game before trying something hazardous, so that you may resume the game, should the results be adverse, without having to repeat a lot of work. As always, use the command "HINT" to decode the hints we publish one issue after each Adventure's appearance. The introduction to each Adventure explains this more fully.

To start up the Adventure, just run the program called "INTRO" or "INTRO.BAS" on your disk, or select the Adventure from the DV menu.

On cassette, the INTRO program is the one just before the Adventure, which is the last program on the tape.

The Adventure runs on any TRS-80 with at least 16K RAM (32K disk).

**Here are the encrypted hints for
Atlantis, the Adventure in issue 41.**

To find Atlantis: ULOOLD GSV NVINZRW.

If something you need seems to be missing: OLLP
RM GSV QFMP LI ZG IVGVK DSVM SV'H
WVZW.

What to do with the coins: GSILD GSVN RMGL
GSV ULFMGZRM.

What to do with the rods: RMHVIG GSVN IZGSVI
GSZM WILK GSVN.

To open the foot locker: WVHGILB RG DRGS GSV
YOZHGVI.

The injured Atlantean: TL YZXP GL DSVIV SV
DZH SFIG ZUGVI BLF'EV SVOKVW SRN.

If you get thrown into jail a second time: TREV FK
--BLF XZM'G VHXXKV UILN QZRO GDRXV!

The Best of SoftSide

☐ **YES...** I want to enjoy the best programs — Games, Utilities, Adventures — from SoftSide Magazine's past. Please rush my copies of The Best of SoftSide to:

Name _____

Address _____

City _____ State _____ Zip _____

Here's my order For The Best of SoftSide

Computer	Quantity Book Only \$19.95 ea.	Quantity Book plus Disk \$68.95 ea.
Apple®		
Atari®		
TRS-80® Mod I		
TRS-80® Mod III		
TOTAL COPIES	@\$19.95	@\$68.95

☐ I've already ordered The Best of SoftSide book. I'm enclosing \$49 for the disks only.

Total Enclosed \$ _____

I'm paying by: ☐ Check ☐ Money Order
☐ VISA ☐ MasterCard

(Foreign order please include \$5 postage. PAY IN USA FUNDS ONLY.)

VISA/MasterCard# and Interbank# _____ Exp. Date _____

Name of Cardholder _____

Signature _____

Like SoftSide? Wait 'til you've seen our BEST!



The Best of SoftSide!

For the past four years, **SoftSide Magazine** has been bringing Apple®, Atari®, and TRS-80® owners the best in BASIC software. But now you can do even better...**The Best of SoftSide**. From all our back issues, we've selected the most useful...the most entertaining...the most fun programs **SoftSide** has ever published.

The Best of SoftSide is available in three versions...one for Apple, Atari, and TRS-80. Each contains page after page of BASIC code for adventures, simulations, practical applications, and much more.

To order your copy of **The Best of SoftSide**, fill out the coupon, and mail it along with \$19.95 to **SoftSide**, 10 Northern Blvd., Northwood Executive Park, Amherst, New Hampshire 03031.



Tired of Typing?

- Use And Enjoy These Programs Right Away!
- Avoid Worry About Typos!
- Get Additional Programs to Enjoy Without Typing.



Order this Issue's Programs on Disk or Cassette!

Send the coupon below to:

SoftSide Publications, Inc.
10 Northern Blvd.
Northwood Executive Park
Amherst, NH 03031



CV DV

SoftSide™ Selections

ORDER THIS ISSUE'S MEDIA

Having difficulty finding the time to input the programs from this issue of *SoftSide Selections*? Don't despair — the cassette and disk versions are still available! And, each DV (disk version) contains an additional program.

☐ #42 DV \$16.00 ☐ #42 CV \$8.00

Disk Only: ☐ Apple® ☐ IBM® PC

Disk or Cassette: ☐ Atari® ☐ TRS-80® Mod. I ☐ TRS-80® Mod. III

Name _____

Address _____

City/State _____ Zip _____

☐ **Check or Money Order**

☐ MasterCard ☐ VISA

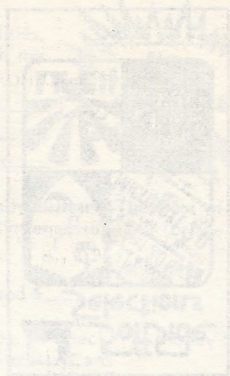
Name of Cardholder _____

MC# or Interbank#/VISA# _____

Exp. Date _____ Signature _____



The Best of SoftSide



General Information

These are the standard procedures for the programs published in **SoftSide Selections**. Sometimes, a particular program does not lend itself to these procedures. Always read the specific instructions accompanying a program. They will instruct you if there are any variances from the following procedures. Also, back issues of **SoftSide** may differ in some details.



SWAT TABLE

At the conclusion of each program listing in **SoftSide Selections**, we include a **SWAT (Strategic Weapon Against Typos)** Table. **SWAT** for the TRS-80 appeared in **SoftSide** Issue #30. If you missed Issue #30, we'll send you a free reprint of **SWAT**. Send a self-addressed, stamped envelope to:

SoftSide Publications, Inc.
Department **SWAT**
10 Northern Blvd.
Northwood Executive Park
Amherst, NH 03031

Please be sure to tell us that you have a TRS-80 computer.

Magnetic Media

Disks are available in Model I or Model III format. They contain the DOSPLUS operating system. A cover program runs automatically when you boot the disk. Back issues earlier than May 1981 are available only in Model I format. If you have a two-drive Model III, you can convert such disks with the CONVERT utility.

Tapes CLOAD in the normal manner on Model I's, and at low speed (500 baud) on Model III's. The first program is a cover/menu program; side two of the tape is a duplicate of side one.

SoftSide Selections disks and tapes are duplicated on reliable, professional equipment. Bad copies are exceedingly rare. Nevertheless, the trip through the mail occasionally results in damage to the sensitive magnetic media. If, after a reasonable number of attempts on well-adjusted, clean equipment, you are unable to load a program, return it to us along with an exact explanation of your problem. We will send you a replacement copy.

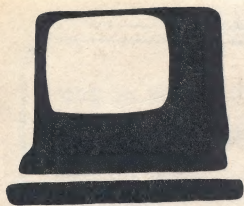
SoftSide Selections media are not copy protected. We urge you to make an archival backup copy of your disk or tape as soon as you receive it, as our replacement policy is valid only for 30 days. Please resist the urge to give away copies of copyrighted material.

Line Listings

The line listings in this booklet are in standard 64-column format, and they appear exactly as they should on your screen when you type LIST.

System Requirements

The necessary memory and other equipment you need to run a program are listed in the introductory paragraph of the article for each program. (Also see the **SoftSide Adventure Series** elsewhere in this booklet.)



SoftSideTM Selections

Here's **SoftSide Selections**, the handy, pull-out booklet with program listings for your TRS-80® Model I or III computer. This issue, **SoftSide Selections** for the TRS-80 features:

- **Microtext 2.0** — This issue's Front Runner is a BASIC word processor with editing, printout, and even search commands.

- **King II** — Your people's welfare (and your crown) depend on your decisions regarding land, crops, and war!

- **TRS-80 DV Bonus Program: Life II** — automates the original game of Life, featured in **Scientific American**, and adds three new twists — animation, sound, and multi-player games.

- **The SoftSide Adventure Series — Arabian Nights.** The daughter of the Caliph of Baghdad lies in a death-like trance, the victim of the evil magician Roxor's spell.

SOFTSIDE PUBLICATIONS, INC.
Amherst, New Hampshire